Context Aware Mobile Application for Mobile Devices

Mfundo Masango*, Francois Mouton†, Alastair Nottingham‡ and Jabu Mtsweni§
Command, Control and Information Warfare
Defence, Peace, Safety and Security
Council for Scientific and Industrial Research
Pretoria, South Africa
*Email: gmasango@csir.co.za
†Email: moutonf@gmail.com
‡Email: anottingham@csir.co.za
§Email: jmtsweni@csir.co.za

***Abstract*—Android smart devices have become an integral part of peoples lives, having evolved beyond the capability of just sending a text message or making a call. Currently, smart devices have applications that can restrict access to other applications on the same device, implemented through user authentication. Android smart devices offer the capability of Android Smart Lock, which uses different authentication methods for unlocking the device based on the users location. However, Android Smart Lock does not allow locking for individual applications. A possible solution to this limitation is an application that performs user authentication using a context-aware approach. This paper proposes a context-aware application, which provides different user authentication methods that are set up according to the auto-detection of areas designated as safe zones by the user. This application aims to improve the overall security of the content of a given device by securing individual applications.**

***Index Terms*—Android; Geofence; Google location services; Lock screen; Pattern lock;**

I. Introduction

Android smart devices such as smart phones and tablets have evolved far beyond the capability of sending text messages or making calls, and are now arcades, personal navigators, storage devices, and social hubs. Another important innovation in modern Android devices is the development of context awareness with respect to the device's given surroundings [1]. Devices are now able to alert users when they are entering or exiting a particular area of interest, and may alert the user by means of a notification or alarm sound [2].

Applications that are context aware are able to track a user's current location, activating different authentication methods based on where the user is situated. These applications make use of geofences; virtual perimeters created around a real-world geographic areas which are dynamically generated [3]. For example, a user may want to be alerted when entering an area that has a McDonald's store, or when they are within the range of a house that they are viewing on a real estate website. In this case, when a user passes the area where the house is located, a notification can be sent to their mobile device that will inform the user of the house's proximity and show the user the house's location via the Global Positioning System (GPS) [2]. The user may then use this information to find the house through a navigation app, such as Google Maps.

GPS receivers are highly accurate in determining a device's exact current location, achieved through the use of satellites [4], [5]. Geofencing is the practice of using GPS or Radio Frequency Identification (RFID) to define a geographic boundary [3]. Once this 'virtual barrier' is established, the administrator can set up triggers that send a text message, email alert, or app notification when a mobile device enters (or exits) the specified area. The location Application Programming Interface (API) available in Google Play services is used for making an application location aware, enabling an application to provide current location tracking and activity recognition [6]. The application is also able to track the current and last location of the user, as well as spoof false geo-locations [6], [7].

Currently, smart devices have applications that restrict access to other applications on the same device. For example, Smart Lock aims to protect a user's device by employing a variety of authentication methods commonly used to unlock a smart device [8]. Some of these methods for authenticating a user include pin, pattern lock, fingerprint, and facial recognition [9]. In general, however, support for protecting applications on an individual basis is limited. The research conducted in this paper proposes adapting some of Android Smart Lock features to protect applications on an individual basis by employing a separate set of user authentication methods. The proposed application has features that allow it to detect a device's current location while also detecting the user's current motion. The application then creates a geofence around a 'safe' location as defined by the user. Geofence transitions will be triggered upon entrance or exit of the virtual perimeter, and uses different authentication methods for unlocking specific applications based on a user's current location and the device's orientation and motion [10].

Securing personal information on a smart device was initially a lengthy process, but has reduced in complexity over

978-1-5090-2473-8/16/$31.00 ©2016 IEEE

time and is now much more straight forward. A user may create a less secure password that may be easy to crack, in which case they may be given a chance to create a more ‘secure’ password. This ‘secure’ password may include a combination of characters, special characters and numerical values, but at the expense of authentication simplicity. Smart Lock was released as a possible solution to this problem at the device level. The application introduced in this paper attempts to solve this problem on a per application basis, providing finer granularity with respect to application security.

The paper is structured as follows. Section 2 gives background on devices adapting to user behaviour, Android Smart Lock and the Fingerprint authentication feature offered in Android 6.0 (Marshmallow). Section 3 describes the Context Aware Mobile Application. Section 4 identifies the application’s features and describes the basic usage of the Context Aware Mobile Application. Section 5 discusses the advantages and limitations of the applications and proposes some scenarios which illustrate the use of the application. Section 6 concludes the paper and discusses potential future work.

II. Background

The following subsections provide a background on: locking options available on Android operating system, Android Smart Lock, Fingerprint authentication — a feature that is introduced on devices with Android 6.0 (Marshmallow) or later, and background on devices adapting to user’s behaviour. This background is provided in order for the reader to familiarise themselves with the current trends within the field of authentication mechanisms within the smart mobile device sphere.

A. Locking on an Android device

Smart devices offer different methods for locking a device, but the most commonly used lockscreen methods are none, swipe, pin, pattern and password [9]. However, some of these methods are not available on all smart devices, with some offering only none or swipe, which does not provide the same level of security as, for example, a pin or password method. Using swipe authentication for example, a user presses the unlock button on the device and with a simple swipe across the screen, the device is unlocked [9].

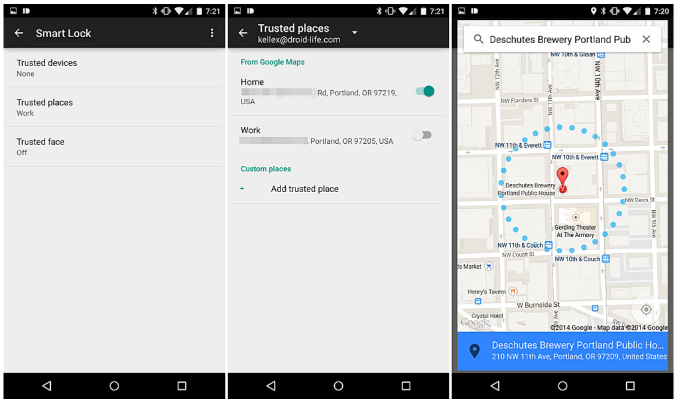
Setting a lockscreen on an Android device is a relatively simple process. A user needs to access security settings on the device and select a lockscreen method. The different methods available are slide, face unlock, pattern, pin and password. It is important to note however that when none of these options are selected, the default lock screen does not require authentication to unlock the device [11]. A pin is a numerical password of 4-to-17 digits, while the password is an alphanumeric string that may contain upper and lower case characters, integers, and special characters. The lock pattern method is a 3-by-3 grid of dots where one can draw straight lines to form a pattern, without visiting a single node more than once [12], [13]. Finally, the face unlock method requires a user to look into the front camera and align within the marked area to perform facial recognition [14].

B. Android Smart Lock

Smart Lock is an Android feature introduced on devices with Android 5.0 (Lollipop) or later [15]. This is a security feature that was designed to allow a user to bypass secure lock screens when a user is near a trusted location. Smart lock allows a user to add a trusted location, expected to be a secure environment such as the user’s home, where authentication is disabled to improve ease of access to the device. Figure 1 shows a user making use of Android Smart Lock.

Depending on a user’s trusted locations, a user may prefer to have quick access to the device, bypassing user authentication. For example, a user may desire a less complicated unlocking mechanism to simplify device access while jogging along a predefined morning route than they typically use in untrusted public locations. Smart Lock will detect the current orientation and current location of the device. When the current location of the device is within a known geofence perimeter, user authentication can be adapted to provide a context appropriate access method to the device.

Facial recognition (or face unlocking) is another new feature that allows a user to unlock their device through the device’s onboard camera [8]. This process is not guaranteed to work perfectly in all instances, and utilises a backup password or pattern lock in case a user’s face is not recognised. One more new feature is body detection, which enables a device to detect when it is being held in the hands of the user or placed in a user’s pocket [16]. The device remains unlocked while in the hands of a previously authenticated user, but resets to standard authentication when the smart device is set down [12].



Smart Lock

Trusted devices
None

Trusted places
Work

Trusted face
Off

Trusted places
kellex@droid-life.com

From Google Maps

Home
[redacted] Rd, Portland, OR 97219, USA

Work
[redacted], Portland, OR 97205, USA

Custom places

+ Add trusted place

Deschutes Brewery Portland Pub

Deschutes Brewery Portland Public Ho...
210 NW 11th Ave, Portland, OR 97209, United States

Fig. 1. Android Smart Lock

C. Fingerprint Authentication

Currently smart devices that operate using Android 6.0 (Marshmallow) have a fingerprint authentication method. The method serves as an alternative for entering a password, using a user’s fingerprint as a means of authentication for gaining access to the device. A user has to create a backup password that will be used to gain access to the device if the user’s

978-1-5090-2473-8/16/$31.00 ©2016 IEEE

fingerprint is not recognised. The user's fingerprint is also used for identification when making purchases in some stores [17], [18].

D. Devices adapting to user's behaviour

User authentication on a device, whether a laptop, mobile device or smart watch, may be done differently depending on the device. A proposed conceptual model [19] allows devices to be part of a centralised authenticity key distributor that is used for verification of a user. Stephen Marsh has proposed a device comfort zone, where an enhanced notion of trust is enabled on a given personal device in order to better regulate the state of interaction between the device, its owner, and the environment [20]. This technology allows devices to gain each other's confidence, allowing them to share the process of user authentication with each other. The devices also secure the smart device's information, neglecting the applications on the smart devices which could be vulnerable to being accessed when all the devices are in proximity of each other. In these cases, a user's behaviour can also be used as a tool for security; for instance, information on the average times that a user accesses certain devices, as well as the current activity of the user on a given device, can contribute to the device better understanding of its user's patterns.

Applications are being developed to learn and adapt to a user's behaviour patterns, tracking the user's current location, heart rate, motion and so forth [21]. These applications are also developed to try and improve the security information on a smart device by providing different authentication methods for accessing the device [22]. Today, many users will own and use a variety of digital devices. For instance, a user may have a smart device, tablet, laptop and a smart watch. These different devices likely have different authentication methods for authenticating a user, and the smart device and tablet may even use the smart watch for user authentication, as these devices may be connected to the smart watch. This allows the devices to communicate with each other, sharing the same identity with each other until the smart watch is not within the area [23]. The devices can still authenticate a user via PIN, password, facial or voice recognition, and a laptop may also have a inbuilt fingerprint scanner for authenticating the current user.

III. CAMA Application

Context Aware Mobile Application (CAMA) is a mobile application that provides different user authentication methods based on device context. The different user authentication methods are triggered by the auto-detection of safe zones. A safe zone indicates a circular area which is parametrised by a geofence, which stores the name, longitude and latitude of the safe zone in a localised database [7], [3]. The first authentication method on the application is a 4 digit password, but different authentication methods can be customised based on the device's context (location, orientation and motion) on a per-application basis, such that the devices current context can be used to select the most appropriate authentication model for gaining access to the application.

The authentication methods are triggered by using Google location services, which allows GPS to track the current location of the device, whether it be via mobile network, GPS or Wi-Fi [24]. When the device is in a safe zone, CAMA will not require any form of authentication to access the requested application. Authentication methods are activated once a user's current location is registered as outside of a safe zone. A saved location's latitude and longitude are used to calculate the distance between the current location of the device and any saved geofences. Locations that are not safe zones will require user authentication via a 4 digit password. When the user is outside fo the perimeter of the geofence, stronger authentication will be required to gain access to the application [25].

CAMA allows a user to register an account, create a user name, a 4 digit password, and a security question to be used for recovery of a forgotten password. Upon the successful creation of a user account, the application will check to ensure that GPS settings are turned on, so as to enable tracking of the user's current location [5]. With the use of Google location services, the current location of a user will be displayed on the provided map. The latitude and longitude of the current location are also displayed. Google time-line is also used by the application to identify areas a user visits regularly [26]. These locations are then suggested to a user as potential safe zones, allowing the user to save them as safe locations. Safe locations that have been identified by the user then have a geofence created around them with a radius of 0.1, as seen in Figure 2.

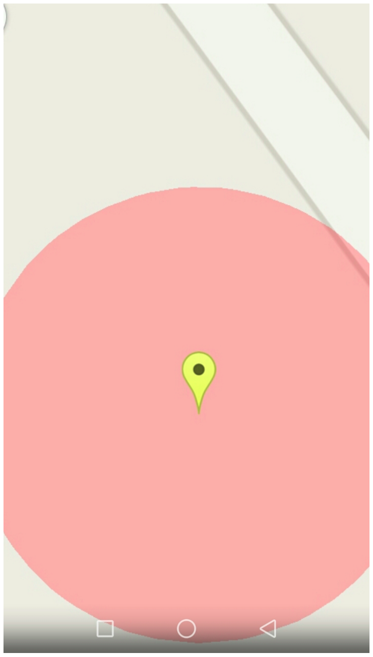


Fig. 2. Geofence with radius of 0.1

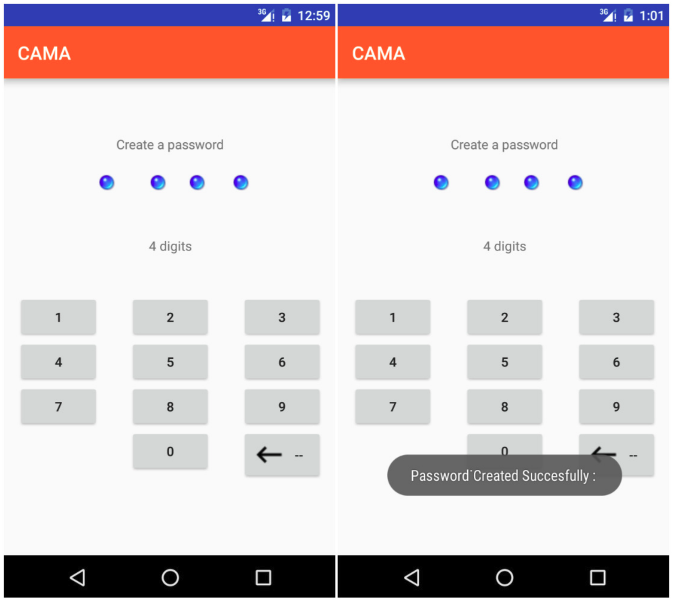
IV. CAMA Usage and Features

A. Registration

A new user will be able to create an account. Upon successful creation of the account a user will be required to create a 4 digit password and select a question, which will be used for user authentication on the mobile application.

B. 4 digit password

A user successfully creates a 4 digit password, as seen in Figure 3. This password will be used for authenticating a user before saving a location, and will also be used to access the application when the authentication method is triggered.



Password Created Succesfully :

Fig. 3. 4 digit password successfully created

C. Security question

A user will be given default security questions such as:

- “Where was I born?”
- “What is my pet’s name?”
- “What is my favourite fruit?”
- “What did I meet my first love?”
- “Where do I attend church?”

The user will select one and enter a answer for the selected question. The question and answer will be used for changing a user’s forgotten password.

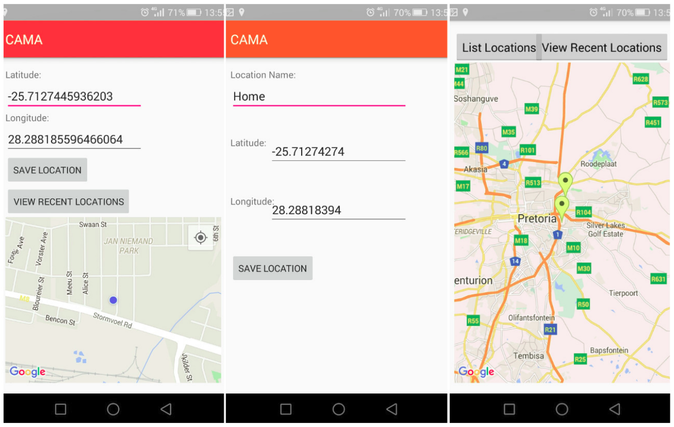
D. Adding a geofence

A user creates a geofence by pressing the ‘save location’ button, at which point the latitude and longitude of the location are displayed for the user. The user will then enter a location name and press the ‘save location’ button, which will create a geofence around that area. The device will then monitor the entering and exiting of the geofence, as it is now registered as a safe zone.

E. Removing a geofence

A list of saved geofences is created and displayed when a user presses the ‘list location’ button; on pressing and holding on a selected location on the list the user will be prompted to select whether the location should be removed as a safe zone.

A user’s current location is viewed and tracked on the map, with longitude and latitude of the exact current location displayed. A user may choose to save their current location as a safe zone via this application. The longitude, latitude and name of the current location are then stored. Locations that have been saved by the user are viewable on a map, as seen in Figure 4.



Latitude: -25.7127445936203

Longitude: 28.288185596466064

Location Name: Home

Latitude: -25.71274274

Longitude: 28.28818394

Fig. 4. Geofence creation

Figure 5 displays different use-cases of the application. The exact current location of the device is tracked and displayed with the current motion of the device. When the exact location of the current device is not within a designated safe zone, user authentication via a 4 digit password is required for accessing the application. The application will keep tracking the device’s current location against any safe zone defined by a user. The following subsections introduce the basic usage of the application.

F. Forgotten password

The question a user selected during registration will be shown, and the pre-determined answer as set by the user must then be entered. Upon the successful submission of an answer, the user will be allowed to create a new 4 digit password.

G. Within a safe zone

Upon entrance of a safe zone a user will receive a notification, indicating to the user that a geofence has been entered and no form of authentication will be required. A user will have ease of access to the application while the device remains within a safe zone.

H. Exit of a safe zone

Upon exit of a safe zone a user will receive a notification, indicating to the user that a geofence has been exited. This will trigger the 4 digit authentication method, requiring the user to enter the 4 digit password before gaining access to the application.

I. Further distance from the safe zone

Based on a user's current location, the application will calculate the distance between the previously entered geofence, and based on that another authentication method will be triggered. The user will need to enter a user name and a password to gain access to the application.

J. Weak GPS Signal

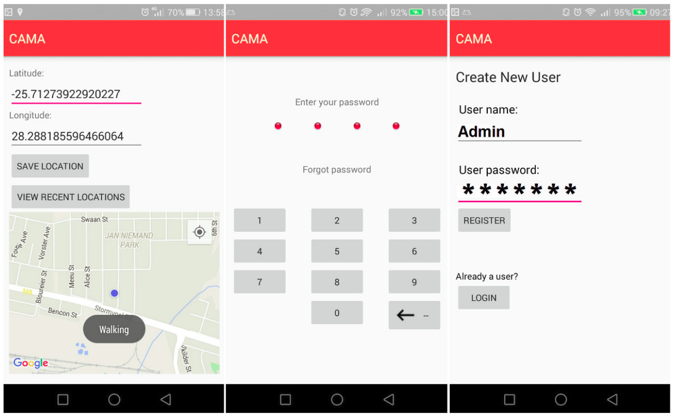
The application will fallback to Wi-Fi and mobile network methods for determining a user's estimated current location. The security level will be a level higher.

K. Not able to determine a location

The application will act as if it is outside of a safe zone and the highest level of security will be immediately activated. This might be the case in buildings.

L. Usage of Google time-line

The CAMA application accesses a user's Google time-line and determines locations that are visited continuously by a user, and suggests these locations to a user as potential safe zones that a user may register in order to have ease of access to the application while in those frequently visited areas. Saving of these locations will be suggested in a timely manner to allow the user to make efficient decisions as to whether the suggested locations can be used as safe zones.



Latitude: -25.71273922920227

Longitude: 28.288185596466064

User name: Admin

User password: *******

Fig. 5. Different uses cases within the CAMA application

V. DISCUSSION

Context aware mobile applications have been explored before, with some being developed to assist users with everyday tasks or with the usage of the mobile device. It may be that a small number of these applications are developed to enhance the security on a smart device, adding an extra security layer beyond those that are pre-installed on the smart device. The CAMA application proposes that an individual application can learn and understand user surroundings, and can therefore provide different user authentication methods for accessing device applications.

The application has some limitations. For example, notifications for entering and exiting of safe zones are not displayed in real-time, and the user must select from a list of default security questions rather than creating their own. The current authentication methods are designed for basic usage of the application, allowing for future development with a front-end user interface. The application's battery consumption is also very high, as the application currently uses fine and coarse locations for tracking the user's current location. This drains the battery as it uses Wi-Fi, GPS and mobile network communications to estimate the exact current location. The application's battery consumption could be reduced by using coarse locations to determine the location, which will allow the application to use Wi-Fi and the mobile network to estimate the user's exact location. With the application having the tracking of transitions as a service running continuously in the background, the entering and exiting of safe zones are monitored continuously. The strength in the application lies in that the application is able to successfully track, in real-time, the location of a user which allows for accurate tracking of entering and exiting of safe zones. This allows the application to switch between the different authentication methods in real-time, thus triggering the different authentication methods.

The proposed application in this paper could be further developed to provide more functionality to the user while still offering ease of access to the application. The following subsections propose scenarios which illustrate real-life use cases for the application.

A. Phone stolen

A user at the office may feel it necessary to secure an individual third party application for security reasons, as the application may have confidential information. If they have chosen to save the office as a safe location, in order to have ease of access to the application, they are still vulnerable to theft, and their device may be taken and accessed by an unauthorised person within the designated safe zone. However, once the device has been taken outside of the designated safe zone, in this case the office, the application will trigger different user authentication methods, locking access to the device.

B. Home

A user is at home and would like to lock his banking application for security purposes, as the user may not want his banking information to be viewed by any unauthorised person. The user may have a specific room in the house saved as a safe zone where the user would like to have ease of access to the banking application. Other rooms in the house will require user authentication before being able to access the application, as they are not safe zones.

C. Gym

A user goes for an afternoon gym session, and may wish to lock his email application for security purposes; the user will then choose not to set the gym's location as a safe zone, and the application will require user authentication. The user may however want quick access to their email application at work and home, and so may choose to save these locations as safe zones, which will then require no user authentication.

These scenarios show that the application being proposed has capabilities beyond tracking a user's current location, providing notifications of entering and exiting safe zones, and saving locations as safe zones.

VI. Conclusion

Smart devices offer different user authentication methods such as slide, face unlock, pattern, pin and password. Today's Android smart devices have implemented more features to improve the security of smart devices and information it contains. User authentication methods have been implemented to ensure restriction of unauthorised users gaining access to the device. CAMA provides different authentication methods on an individual application and uses procedures that differ from Smart Lock, allowing the application to be self aware and responsible for its own user authentication methods.

The CAMA application is a context aware application that will track the user's current location, current motion and allow a user to save locations. The application makes use of the user's Google time-line and auto save locations that are recognised as consistently visited locations, and suggests locations for the user to save as safe zones. Locations that have been created have geofences around them which are used to provide different user authentication methods for gaining access to device applications. Specific applications can be individually secured and provide different user authentications methods depending on device context.

Future work will be done to store more information on the application such that a user may securely store emergency contact information, media files and documentation. Further development of the application will allow the application to provide more user authentication methods. This will allow a user to enter their own security question, as well as the implementation of other actions beyond saving locations and detecting the current motion of a user's device. For future avenues of research, it would be useful for a user to be able to integrate the authentication features of CAMA with other applications.

References

[1] C. W. Thompson, "Smart devices and soft controllers," *IEEE Internet Computing*, vol. 9, no. 1, pp. 82–85, Jan 2005.

[2] L. Chamberlain. (2016, March) What is geofencing? GeoMarketing. [Online]. Available: http://www.geomarketing.com/geomarketing-101-what-is-geofencing

[3] S. Rodriguez Garzon and B. Deva, "Geofencing 2.0: taking location-based notifications to the next level," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*. New York, NY, USA: ACM, 2014, pp. 921–932.

[4] Garmin. (2016, April) What is gps? Garmin. [Online]. Available: http://www8.garmin.com/aboutGPS/

[5] M. Avdyushkin and M. Rahman, "Secure location validation with wi-fi geo-fencing and nfc," in *Trustcom/BigDataSE/ISPA, 2015 IEEE*, vol. 1. IEEE, Aug 2015, pp. 890–896.

[6] Google Developers. (2016, April) Using google api. Google. [Online]. Available: http://www.developer.android.com/

[7] A. Popescu. (2013, October) Geolocation api specification. Google, Inc. [Online]. Available: https://www.w3.org/TR/geolocation-API/

[8] Kellex. (2014, Nov) Android 5.0 feature: Google updates smart lock on lollipop to include trusted places. Droidlife. [Online]. Available: http://www.droid-life.com/2014/11/18/android-5-0-feature-google-updates-smart-lock-o

[9] T. Sixta, "Gesture recognition for mobile phone unlocking," Master's thesis, Czech Technical University in Prague, Center for Machine Perception, Department of Cybernetics, Faculty of Electrical Engineering, Czech Technical University, May 2014.

[10] J. R. Kwapisz, G. M. Weiss, and S. A. Moore, "Activity recognition using cell phone accelerometers," *SIGKDD Explor. Newsl.*, vol. 12, no. 2, pp. 74–82, Mar. 2011. [Online]. Available: http://doi.acm.org/10.1145/1964897.1964918

[11] R. B. Malhotra, A. A. Fulzele, and R. N. Verma, "A novel approach for android security system," *International Journal of Computer Engineering and Applications*, vol. 1, no. 1, January 2016.

[12] G. Mazo. (2012, july) How to set up face unlock on your android phone. Androidcentral. [Online]. Available: http://www.androidcentral.com/how-set-face-unlock-your-htc-one-x-or-evo-4g-lte

[13] Q. Kennemer. (2014, March) How to setup a lock-screen pattern, pin or password on your android device [android 101]. Phandroid. [Online]. Available: http://phandroid.com/2014/03/20/android-101-lock-screen/

[14] S. A. A. K. Oka, P. I. K. G. Darma, and A. Arismandika, "Face recognition system on android using eigenface method," *Journal of Theoretical & Applied Information Technology*, vol. 61, no. 1, pp. 128 – 134, 2014.

[15] N. Elenkov. (2014, dec) Dissecting lollipops smart lock. Google. [Online]. Available: http://www.developer.android.com/

[16] J. Duino. (2015, aug) On-body detection explained. Androidcentral. [Online]. Available: http://www.androidcentral.com/body-detection-explained

[17] B. Cha, K. Kim, and H. Na, "Random password generation of otp system using changed location and angle of fingerprint features," in *Computer and Information Technology, 2008. CIT 2008. 8th IEEE International Conference on*, July 2008, pp. 420–425.

[18] A. De Luca, A. Hang, F. Brudy, C. Lindner, and H. Hussmann, "Touch me once and i know it's you!: Implicit authentication based on touch screen patterns," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, ser. CHI '12. New York, NY, USA: ACM, 2012, pp. 987–996. [Online]. Available: http://doi.acm.org/10.1145/2207676.2208544

[19] A. Al Abdulwahid, N. Clarke, S. Furnell, and I. Stengel, "A concepstual model for federated authentication in the cloud," in *Proceedings of the 11th Australian Information Security Management Conference*. SRI Security Research Institute, Edith Cowan University, Perth, Western Australia, 2013, pp. 1 – 11.

[20] S. Marsh, Y. Wang, S. Noël, L. Robart, and J. Stewart, "Device comfort for mobile health information accessibility," in *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*. IEEE, 2013, pp. 377–380.

[21] M. Mitchell, "Context and bio-aware mobile applications," Master's thesis, Florida State University, 2011.

[22] S. Marsh, Y. Wang, S. Nol, L. Robart, and J. Stewart, "Device comfort for mobile health information accessibility," in *Privacy, Security and Trust (PST), 2013 Eleventh Annual International Conference on*, July 2013, pp. 377–380.

[23] S. Davidson, D. Smith, C. Yang, and S. Cheah, *Smartwatch User Identification as a Means of Authentication*, Department of Computer Science and Engineering Std., 2016.

[24] P. J. Brown, J. D. Bovey, and X. Chen, "Context-aware applications: from the laboratory to the marketplace," *IEEE Personal Communications*, vol. 4, no. 5, pp. 58–64, Oct 1997.

[25] M. Heinl, "Android security," Master's thesis, Offenburg University of Applied Sciences, June 2015.

[26] M. Knoll. (2016, April) Can't remember last night? google's location history can tell where you were. Trendblog. [Online]. Available: http://trendblog.net/cant-remember-last-night-google-location-history-can-help-you/